Development team

**Stefan Crigler**  PCB design, software help, team lead

**Robert Tremewan**  PCB design, software help

**Renny Hong**  software design, RF/DSP

**Arthur Lobins**  software design, RF/DSP

**Cynthia Alvarez-Preciado**  software design, Fusion/UI

**Eric Nystrom**  CACI Mentors

**Chris Chan**

**Jeff Longo**

**James Cook**
Quick refresher

- Locate Bluetooth devices using XTRX software defined radios (SDR) programmed with direction-finding algorithms
- Display current location and location history on Android app in an easy-to-use interface
- Building on last year’s capstone project BlueDentist, which captured Bluetooth packet information
What has hardware team been doing?
PCB revision 1

What we started with.
PCB revision 1

Ethernet and HDMI
PCB revision 1

Mini PCIe for XTRX SDR
BlueFinder PCB
Mini PCIe #1

- Jetson has 5 PCIe device controllers
- A clock from a different controller was connected to the first XTRX
BlueFinder PCB
Mini PCIe #1

- Reroute the length matched differential pairs
BlueFinder PCB

- Only one HDMI worked, so removed second HDMI for space optimization
BlueFinder PCB

- Attempted to fix ethernet
- Checked schematics, firmware, PCB trace against specifications and reference designs
- Probed board to find misbehaving signals
- Large part that is not necessary for our data transmission method
- Removed from board to allow for optimal second mini PCIe placement
Mini PCIe background

- Serial interface standard for connecting high speed components
- Smaller version of PCIe usually used for laptop network cards
- Width (number of lanes): x1, x2, x4, x8, x16 and x32
  - Number of lanes equates to number of bits per cycle that can be moved
  - More lanes = higher bandwidth and transfer rate
- The standard implementation of Mini PCIe peripherals have both x1 PCIe and USB 2.0 compatibility
  - XTRX is slightly different pinout, w/ an additional PCIe lane
BlueFinder PCB: Adding a new mini PCIe

1. Find a free PCIe controller and associated set of lanes, make sure they can be used as 2x PCIe
2. Create schematics
3. Place components on PCB layout file
4. Trace signals
5. Check work against constraints
   a. Matched lengths for differential pairs
   b. Impedance matching and via limits
   c. Location relative to other signals
6. Move other traces and polygons as necessary to abide by constraints
BlueFinder PCB

Adding new mini PCIe

Power routing had to be redesigned to make room for the new high-speed traces
BlueFinder PCB

- Last step is to make the PCB more compact
- Rearrange layout so the right quarter can be removed
Revisions compared

OLD

NEW
Revisions compared

OLD

NEW
What has software team been doing?
Review of Software Flow
Review of Software Flow
2nd Antenna

- The BlueDentist project only needed one antenna, but two are needed to perform direction finding for BlueFinder.

- We are currently working on adding second frame to account for the second antenna's captured data.

- Both frames will be used in the calculation for direction finding.
Data Collection

- Monitoring the 2.4GHz band for interesting bursts of signals
- The data is processed if it contains a Bluetooth signal under a certain noise threshold
- Interesting frames are decoded and used in our direction finding algorithm
Angle of Arrival Algorithm

- The angle of arrival algorithm was first written in MatLab
- Algorithm was tested with simulated MatLab data that was made to be similar to expected real data (included noise estimations). This worked in Matlab.
- Algorithm was then translated into Rust.
- Currently the algorithm works in Rust using the same MatLab data!
MUSIC Algorithm

- Estimate the autocorrelation matrix using an eigenspace method
  1. Calculate sample covariance matrix
  2. Eigendecomposition
    - Largest eigenvalues and corresponding eigenvectors span the signal subspace
    - The rest corresponds to noise space
- Signal vectors in the signal subspace must be orthogonal to the noise space
  3. Generate complex sinusoids of various incident angles
  4. Measure the level of orthogonality w.r.t. noise space
- Angle resulting in the highest orthogonality is the direction of the signal source
Angle of Arrival Algorithm Demo
Achievements

Fall quarter

Learned Altium

Fixed mini PCIe clock port traces

Learned about current revision from the designers

Began researching ethernet, HDMI specifications and comparing against current revision

Learned CUDA

 Became acquainted with BlueDentist software flow

Gathered and built the software from last quarter

Implemented Angle of Arrival (AoA) algorithm in Matlab

Winter quarter

Added a second mini PCIe for the second XTRX SDR

Moved signals and power planes to optimize space and allow for second mini PCIe

Updated footprints for components

Attempted Ethernet fix

Bill of materials updated, component sourcing verified

Learned Rust

Implemented the second antenna

Modified data processing to include a second data frame

Translated AoA program to Rust
What more needs to be done?
Spring Quarter Goals

**HW**
- Send in designs for manufacturing
- Modify firmware in line with changed and added signals from the SOM
- Debug and finalize PCB

**SW**
- Integrate the software to work on the custom hardware
- Implement time difference of arrival (TDoA)
- Expand software to track additional devices
- Improve decode thread to decrypt Bluetooth headers
- Make an android app that displays current locations and history
Acknowledgements